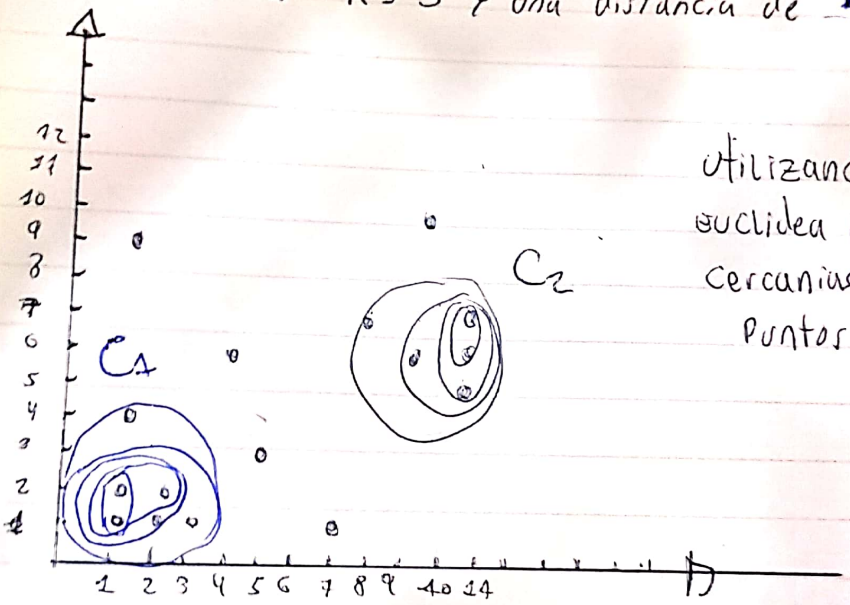
SAN MARTIN Nicolas
104320

1) Para solucionar este problema y poderlo visualizar los puntos, utilizaría DBSCAN para que aquellos puntos en un estado intermedio, que los considero "ruido" queden sin clasificar.

Para eso defino los hiperparametros de DBSCAN que son K y ε.

Voy a tomar K = 3 y una distancia de $\eta = \varepsilon$



utilizando la dist. euclidea para evaluar cercanias entre puntos y clusters

Voy a unir los puntos que están a una distancia menor a ε=(2) de K = 3 puntos

De esta forma podría solucionarse para este caso esos puntos que no serían de ni un cluster o de otro, aunque como vemos, el punto (9,10) debería ir al Cluster 2 pero con la distancia que use no entra, se podría buscar una que sea más optima.

No se si entendí bien lo que pedía el ejercicio, en caso de que esos puntos "ruidosos" tengan que ir a algun grupo sí o sí, usaría ... sigue..

un algoritmo como K-means ++ o clustering Jerarquico para buscar nuevos centroides y que se ajuste a los puntos pedidos.

Anoto a cuantos nodos linkea la Pu y Pi

$Q = \{ P_0 = 2,\ P_2 = 1,\ P_4 = 2,\ P_6 = 1,\ P_1 = 1,\ P_3 = 4,\ P_5 = 1,\ P_8 = 1 \}$

Prob de teletr. a cualq

3- 1) Inicializando los pesos de cada nodo en 1/8:

Lo que hago para cada pagina es, primero multiplicar β x el peso del nodo y linkea a esa pagina, multiplicado por la probabilidad que le corresponde repartir teniendo en cuenta a las paginas a las cual le da peso y sumandole $(1-\beta)\cdot\frac{1}{8}$ que es la prob de teletransportarse a cualquier nodo.

β · Peso inicial · no recibe de nadie · $1-\beta = \frac{1}{5}$

$P0 = 0.8 \cdot \left(\frac{1}{5} \cdot (0)\right) + 0.2 \cdot 1/8 = 0.025$

$P1 = 0.8 \cdot (1/8 \cdot ((1) + 1/2)) + 0.2 \cdot 1/8 = 7/40$

$P2 = 0.8 \cdot (1/8 \cdot (1/2 + 1/4)) + 0.2 \cdot 1/8 = 1/10$

$P3 = 0.8 \cdot (1/8 \cdot (1)) + 0.2 \cdot 1/8 = 1/8$

$P4 = 0.8 \cdot (1/8 \cdot (1 + 1/4)) + 0.2 \cdot 1/8 = 3/20$

$P5 = 0.8 \cdot (1/8 \cdot (1/4 + 1/2) + 0.2 \cdot 1/8 = 1/20$

$P6 = 0.8 \cdot (1/8 \cdot (1/4)) + 0.2 \cdot 1/8 = 1/20$

$P8 = 0.8 \cdot (1/8 \cdot (\underbrace{1/2}_{\text{nodo 4}} + \underbrace{1}_{\text{nodo 5}} + \underbrace{1}_{\text{nodo 6}})) + 0.2 \cdot 1/8 = 0.275$

Vuelvo a iterar

$P0 = 0.8 \cdot (\underbrace{0}_{\text{no recibe de nadie}}) + 0.2 \cdot 1/8 = 0.025$

$P1 = 0.8 \cdot (1/10 \cdot 1/2) + 0.2 \cdot 1/8 = 0.065$

$P2 = 0.8 \cdot (1/8 \cdot 1/4 + 0.025 \cdot 1/2) + 0.2 \cdot 1/8) = 3/50$

$P3 = 0.8 \cdot (0.275 \cdot 1) + 0.2 \cdot 1/8 = 49/200$

$P4 = 0.8 \cdot (1 \cdot 0.025 + 1/4 \cdot 1/8) + 0.2 \cdot 1/8 = 7/100$

$P5 = 0.8 \cdot (1/2 \cdot 3/20 + 1/8 \cdot 1/4) + 0.2 \cdot 1/8 = 11/100$

$P6 = 0.8 \cdot (1/8 \cdot 1/4) + 0.2 \cdot 1/8 = 1/20$

$P8 = 0.8 \cdot (1/2 \cdot 3/20 + 1/10 \cdot 1 + 1 \cdot 1/20) + 0.2 \cdot 1/8 = 41/2$

No sigo iterando porque lo estoy haciendo muy lento

B) Detectar que B no es confiable supondria cambiar la proba de teletransportarse (en este caso era 1/8) a 0 para el nodo 0 y sumarle a cada demas nodos $\frac{1-\beta}{|P|}$ ↗ cant. Paginas confiables.

Por lo que al hacer esto y el nodo Page 0 no estar linkeado por ninguna otra pagina, su peso quedaria en 0 y nunca se llegaria a el.

4) Stream: $S = \{3, 6, 6, 3, 3, 6, 5, 6\}$
$h(x) = x \bmod 32$.

A) Hasheando cada elemento:

$h(3) = 3 \rightarrow 00011$
$h(6) = 6 \rightarrow 00110$
$h(5) = 5 \rightarrow 00101$

como flajolet martin cuenta los 0, en este caso a derecha, procesando el stream resulta
Sea r el contador de 0 a derecha

3 → 00011	r = 0
6 → 00110	r = 1
6 → 00110	r = 1
3 → 00011	r = 0
3 → 00011	r = 1
6 → 00110	r = 1
5 → 00101	r = 1
6 → 00110	r = 1

Como en FM la cant. elementos dif (orden 0) es $2^r = 2^1 = 2$, está mal pory el M°(5) real es 3.

B) Una solución podría ser $h(x) = x + 5 \bmod 32$ $a=1, b=6$
$h(3) = 01001$
$h(5) = 01011$ Donde los contadores
$h(6) = 01100$ quedarían en r = 2 al procesar el 6 y por lo tanto $2^2 = 4$, que aunque bien sigue estimando 1 y un
de mal

Posible solución efectiva para este stream

b) Lo mejor sería colocar otra función de Hashing más con otro estimador para que en Momento sea estimado como $z^2 = 4$ y el otro $z^1 = 2$. Para que al calcular la media de 3 y estime correctamente

c) Propongo $a = 2$ y $b = 3$

$$h(x) = 2x + 3 \mod 32$$

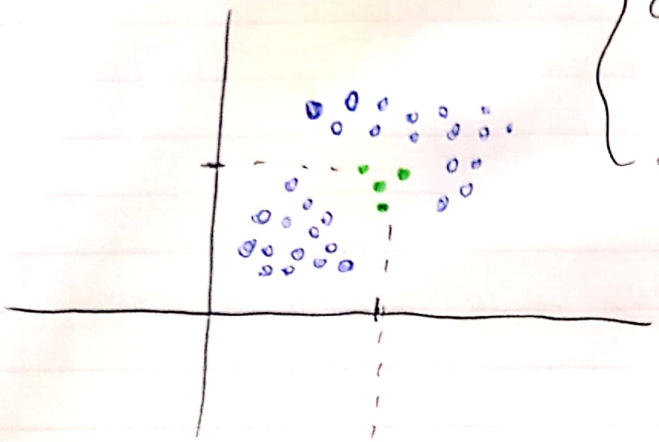
$h(3) = 01001$
$h(5) = 01101$
$h(6) = 01111$

que como se puede ver, si seguimos procesando el stream el contador quedará en 0 y $M^0(5) = 2^r = 2^0 = 1$

No funciona

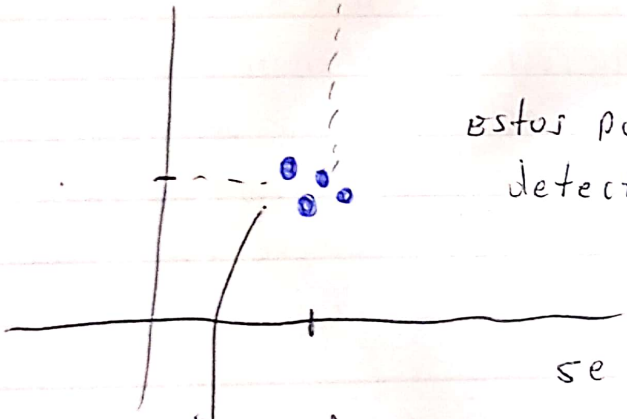
d) KNN con K = 2

Si tengo un modelo en el plano que entrena con los siguientes puntos:



con un set de entrenamiento Desbalanceado con mayoría de puntos de clase azul.

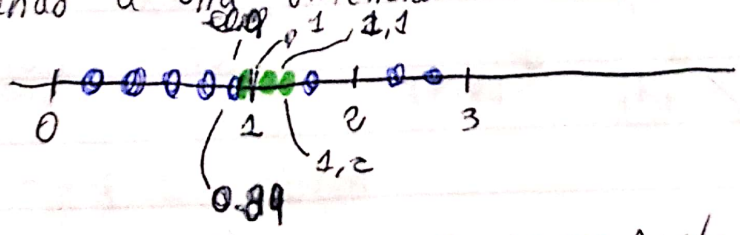
y al testearlo se utilizan los sig. puntos:



estos puntos azules los detectaría como **verdes** debido a que el modelo está overfitteando se podría solucionar con un K más grande

☆ Clasificaría mal porque los compararía con la dist mas corta de los q tiene en su entrenamiento

trasladando a una dimensión



cualquier punto que sea de clase azul pero esté entre el "ruido" de 1 y 1,2 tomará los 2 puntos mas cercanos q serán 1 y 1,1 o 1,2 y lo colocará como verde.

SAN MARTIN Hoja 2/5
NICOLAS 104320

2) Como necesito usar user-user para el usuario a, tomo los usuarios que hayan calificado la pelicula 3 y comparo sus similitudes con a, estos usuarios son: c, e, f.

a: [1 ? ? 5 1 ? 2 3]
c: [2 2 4 ? 2 ? 4 2]
e: [? 3 3 ? 1 4 1 2]
f: [4 ? 5 1 ? ? 4 1]

Pearson es restar el promedio de cada vector y Normalizarlo

Ahora tomo solo las componentes en comun con a

a = [1 ? 5 1 2 3] , Prom = 2.4
c = [2 4 ? 2 4 2] , Prom = 2.8
e = [? 3 ? 1 1 2] , Prom = 1.75
f = [4 5 1 ? 4 1] , Prom = 3

Resto el Promedio a cada vector

a = [-7/5, ?, 13/5, -7/5, -2/5, 0.6]
c = [-4/5, 1.2, ?, -4/5, 1.2, -4/5]
e = [?, 1.25, ?, -0.75, -0.75, 0.25]
f = [1, 2, -2, ?, 1, -2]

Normalizo vectores:

a = [-0.418, ?, 0.7768, -0.418, -0.119, 0.179]
c = [-0.365, 0.5477, ?, -0.365, 0.5477, -0.365]
e = [?, 0.753, ?, -0.452, -0.4522, 0.15]
f = [0.267, 0.534, -0.534, ?, 0.267, -0.5345]

Los reescribo:

$$a = [-0.41,\ ?,\ 0.776,\ -0.418,\ -0.11,\ 0.179]$$
$$c = [-0.36,\ 0.54,\ 1,\ -0.365,\ 0.54,\ -0.365]$$
$$e = [?,\ 0.75,\ ?,\ -0.45,\ -0.45,\ 0.15]$$
$$f = [0.267,\ 0.53,\ -0.53,\ ?,\ 0.267,\ -0.53]$$

Ahora busco las similaridades entre ellos haciendo el producto de vectores:

sim (a, c) = 0.175
sim (a, e) = 0.165
sim (a, f) = -0.64

} tomo estos 2 ya que el otro es negativo

→ calculo el rating

$$\Rightarrow r_{a,3} = \frac{0.175 \cdot (4) + 0.165 \cdot (3)}{0.175 + 0.165} \simeq 3.51 \simeq (4)$$

(4 = valoración del user c; 3 = valoración user e)

lo cual tiene sentido ya que la de los otros eran 4 y 3 para esa peli.

5)

b) usando 2 modelos de Regresión softmax con 2 clases (lo que sería Regresión Logística)

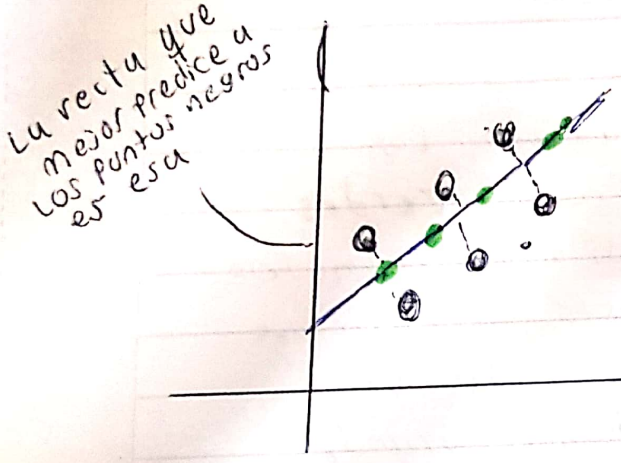
a) Regresión: usando un modelo de Regresión Lineal. sean los

puntos para train y test

el modelo entrenó

$h(x) = Y = mx + b$

sea la recta de entrenamiento que está a igual distancia de los puntos negros de cada lado (lo grafiqué mal pero deberían estar a igual distancia)

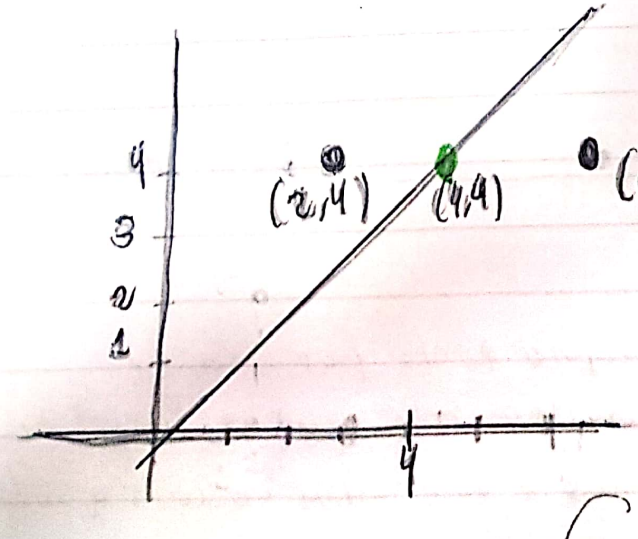


Los puntos verdes son con los cuales se entrenó, que pasan justo por la recta con lo cual no tienen error

↓ ej numerico

$Y = X$

puntos: (2,4), (6,4)



$$MSE = \frac{1}{n} \sum (\bar{x} - x)^2$$

$\bar{x}$: valor real, x: valor que predice

Para este caso de estos 2 puntos es 0